Amendments to the Specification:

Replace paragraph [0001] of the specification with the following:

[0001] This application is related by subject matter to the inventions disclosed in the following commonly assigned applications: U.S. Patent Application No. 10/647,058 ~~(not yet assigned) (Atty. Docket No. MSFT-1748)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR REPRESENTING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM BUT INDEPENDENT OF PHYSICAL REPRESENTATION"; U.S. Patent Application No. 10/646,941 ~~(not yet assigned) (Atty. Docket No. MSFT-1749)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR SEPARATING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM FROM THEIR PHYSICAL ORGANIZATION"; U.S. Patent Application No. 10/646,940 ~~(not yet assigned) (Atty. Docket No. MSFT-1750)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR THE IMPLEMENTATION OF A BASE SCHEMA FOR ORGANIZING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM"; U.S. Patent Application No. 10/646,632 ~~(not yet assigned) (Atty. Docket No. MSFT-1751)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR THE IMPLEMENTATION OF A CORE SCHEMA FOR PROVIDING A TOP-LEVEL STRUCTURE FOR ORGANIZING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM"; U.S. Patent Application No. 10/646,575 ~~(not yet assigned) (Atty. Docket No. MSFT-2733)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR INTERFACING APPLICATION PROGRAMS WITH AN ITEM-BASED STORAGE PLATFORM"; U.S. Patent Application No. 10/646,646 ~~(not yet assigned) (Atty. Docket No. MSFT-2734)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "STORAGE PLATFORM FOR ORGANIZING, SEARCHING, AND SHARING DATA"; and U.S. Patent Application No. 10/646,580 ~~(not yet assigned) (Atty. Docket No. MSFT-2735)~~, filed August 21, 2003 ~~on even date herewith~~, entitled "SYSTEMS AND METHODS FOR DATA MODELING IN AN ITEM-BASED STORAGE PLATFORM".

Please replace paragraph [0055] of the specification and the material immediately below the paragraph as follows:

[0055] Fig. 29 is a diagram illustrating an exemplary Item hierarchy in accordance with an embodiment of the present invention.

~~*[Remainder of Page Intentionally Left Blank]*~~

Please replace paragraph [0583] of the specification and the material immediately below the paragraph as follows:

[0583] As is apparent from the above, all or portions of the various systems, methods, and aspects of the present invention may be embodied in the form of program code (i.e., instructions). This program code may be stored on a computer-readable medium, such as a magnetic, electrical, or optical storage medium, including without limitation a floppy diskette, CD-ROM, CD-RW, DVD-ROM, DVD-RAM, magnetic tape, flash memory, hard disk drive, or any other machine-readable storage medium, wherein, when the program code is loaded into and executed by a machine, such as a computer or server, the machine becomes an apparatus for practicing the invention. The present invention may also be embodied in the form of program code that is transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, over a network, including the Internet or an intranet, or via any other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the invention. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique apparatus that operates analogously to specific logic circuits.

~~*[Remainder of Page Intentionally Left Blank]*~~

Please replace page 184 of the specification as follows:

```
// Indicates that objects from the identity map should be returned. If the content
   of
```

```
    // an object has been modified by the application, the modified object's content
       is
    // preserved. If the object has not been modified, its content is updated with the
    // data returned by the search. The Application may provide an handler for the
    // SearchCollision event and selectivly update the object as desired.
    PreserveModifiedObjects,

    // Indicates that the objects from the identity map should be returned. The
       content
    // of the object is updated with the data returned by the search, even if the
       object
    // has been modified by the application. If this option is specified the Search-
    // Collision event will not be raised.
    OverwriteModifiedObjects
  }

  // The current update operation.
  public enum UpdateOperation
  {
    // Provided when Update is first called. CurrentObject will be null.
    OverallUpdateStarting,

    // Provided just before Update returns after a successful update. CurrentObject
       will be
    // null.
    OverallUpdateCompletedSucessfully,

    // Provided just before Update throws an exception. CurrentObject will be the
       exception
    // object.
    OverallUpdateCompletedUnsuccessfully,

    // Provided when the update of an object is started. CurrentObject will reference
       the
    // object that will be used for the updated.
    ObjectUpdateStaring,

    // Provided when a new connection is needed. CurrentObject will be a string
       that contains
    // the path identifying an item domain as passed to ItemContext.Open or
       retrieved from
    // the Location field of a relationship.
    OpeningConnection
  }
}
```

~~*[Remainder of Page Intentionally Left Blank]*~~

Please replace page 191 of the specification as follows:

```
{
 // A common interface for iterating over objects.
 public interface IObjectReader : IEnumerable, IDisposable
 {

  object Current {get;}
  bool IsClosed {get;}
  bool HasResults {get;}
  Type ObjectType {get;}

  bool Read();
  void Close();

 }

 // Adds asynchronous methods to IObjectReader
 public interface IAsyncObjectReader : IObjectReader
 {

  IAsyncResult BeginRead( AsyncCallback callback, object state );
  bool EndRead( IAsyncResult result );

 }
}
```

~~*[Remainder of Page Intentionally Left Blank]*~~